Institute for Cognitive Systems
TUM Department of Electrical and Computer Engineering
Technical University of Munich

Opponent Detection and Localization

Humanoid Robocup Introduction/Advanced Lab

Bowen Ma

Siyu Chen

Yixi Zhao

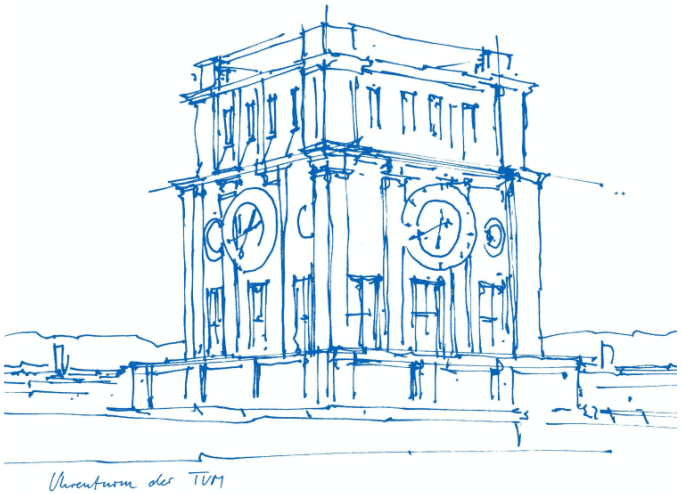
Zhen Zhou

Supervisor : M.Sc. Quentin Leboutet

Chair for Cognitve Systems

Prof. Dr. Gordon Cheng

Munich, XX. March 2019



Motivation

- Lack of opponent detection
- No fixed features in robotics (change as different state)
- Can be applied in other fields (like ball detection)

Approach

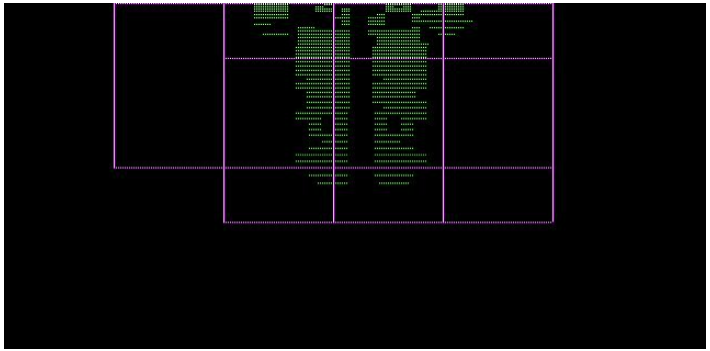
- More reliable preprocessing
- CNN-based approach
- Open-CV approach

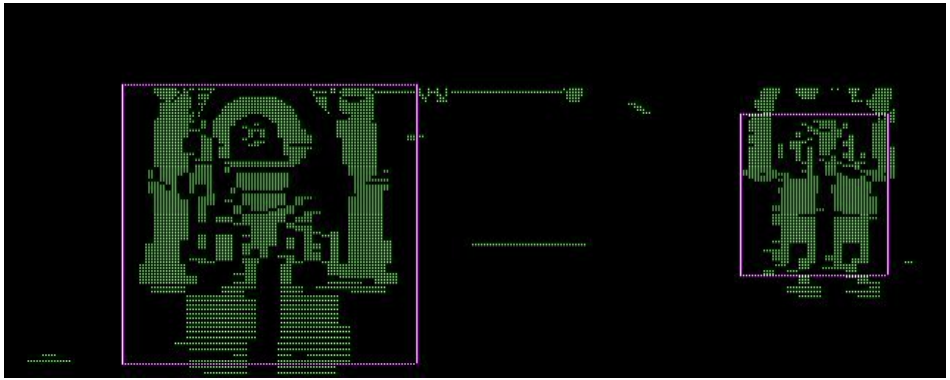
Institute for Cognitive Systems
TUM Department of Electrical and Computer Engineering
Technical University of Munich

Preprocess Image

Approach:

1. Create filter image
   pixel between edges = 1
1. Use sliding window
   a. Shift by 30 pixel
   b. If sum > threshold
      →valid window
1. Combine valid windows
2. Cut window image



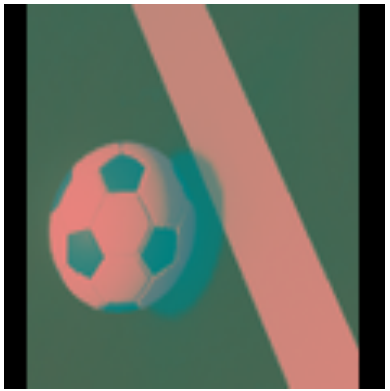


Institute for Cognitive Systems
TUM Department of Electrical and Computer Engineering
Technical University of Munich

Dataset

- Use the dataset from https://github.com/szemenyeim/ROBO
- Select image with robot after preprocessing

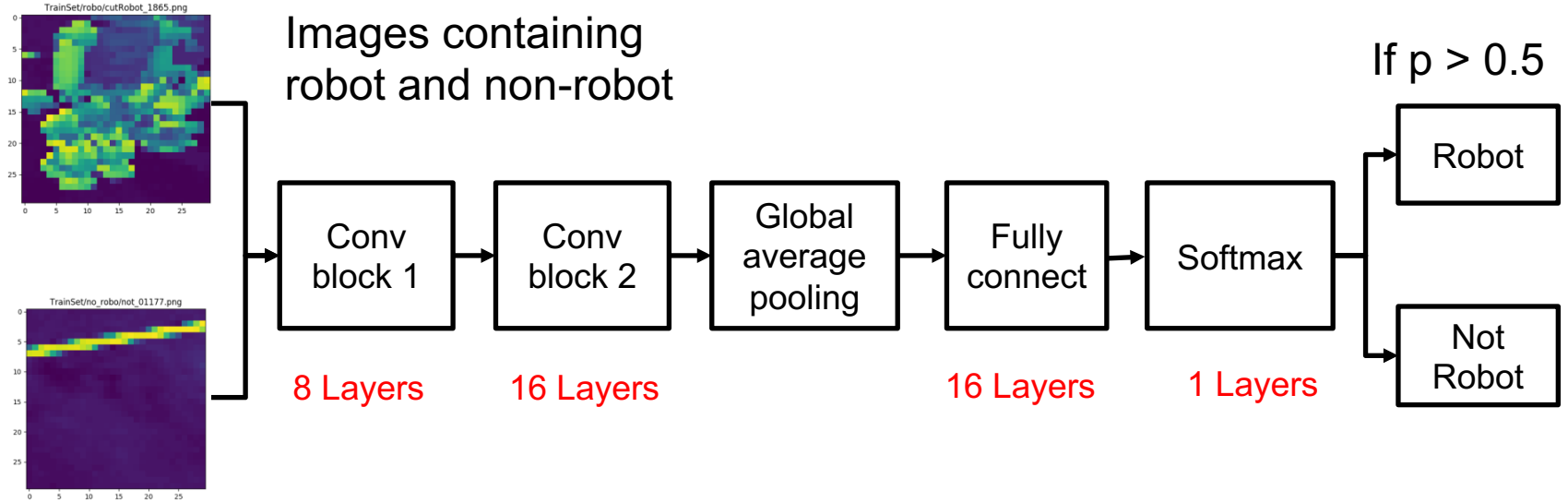
Not Robot : 2016 Pictures



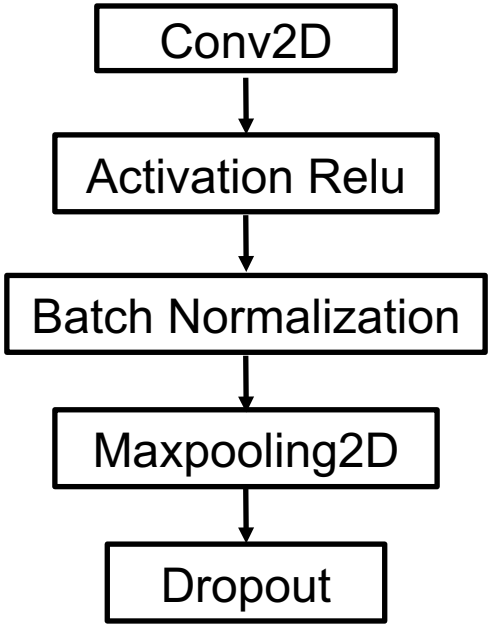
Robot : 2233 Pictures

Institute for Cognitive Systems
TUM Department of Electrical and Computer Engineering
Technical University of Munich

CNN Architecture



CNN Block


Conv2D
Activation Relu
Batch Normalization
Maxpooling2D
Dropout

CNN for Binary Classification

Hyperparameters

- CNN filters: [4, 8], [8, 16], [8, 16, 32], [16, 32, 64], [32, 64, 128]
- FC filters: [8, 1], [16, 1], [32, 1], [64, 1], [128, 1]
- Optimizer: Adam (lr=1e-4, decay=1e-3)
- Loss: Binary cross entropy
- Metrics: Accuracy

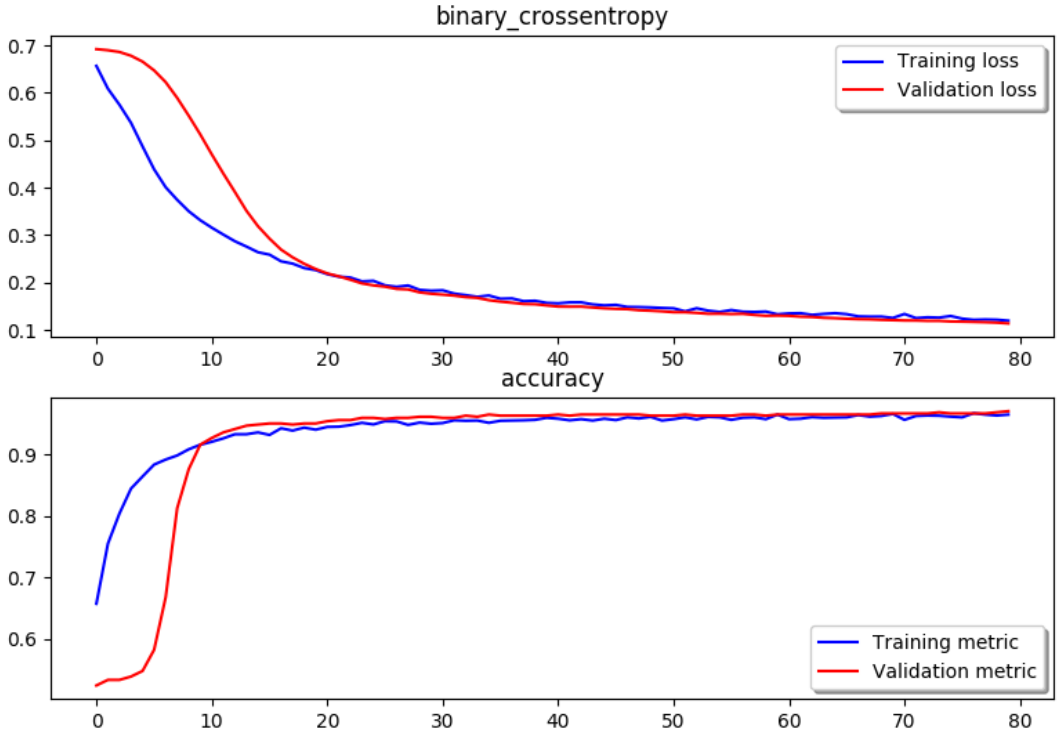
→ Best filter size: [8, 16] [16, 1]

Training Result

Loss:
Training = 0.1197
Validation = 0.1140

Accuracy:
Training = 0.9654
Validation = 0.9710

Test = 0.9738



Test on Real Robot

Time consumption:

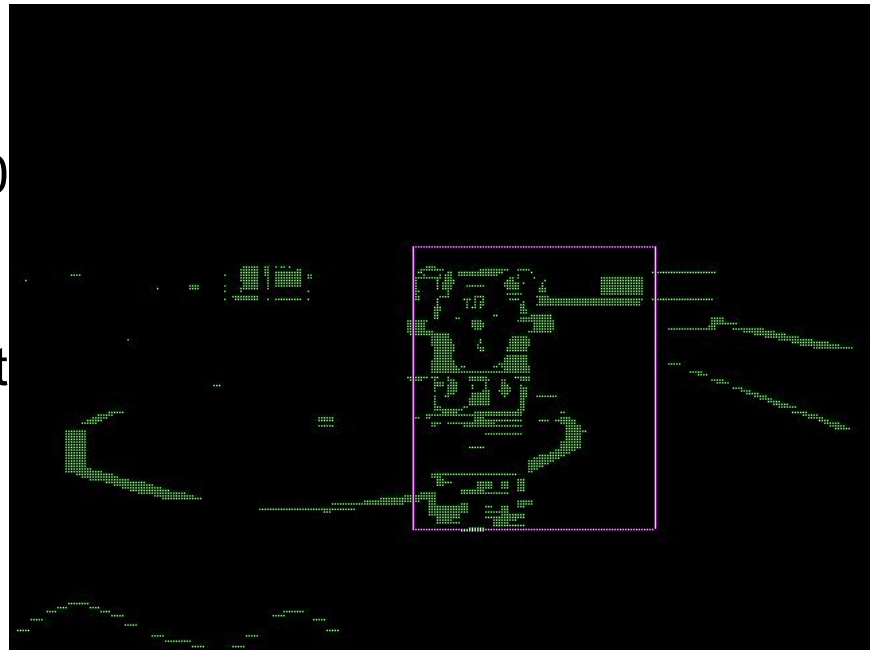
Max Preprocessing = 0.00s

Max Robot detection = 0.0

Problems:

Edge detection not perfect

Light fluctuation

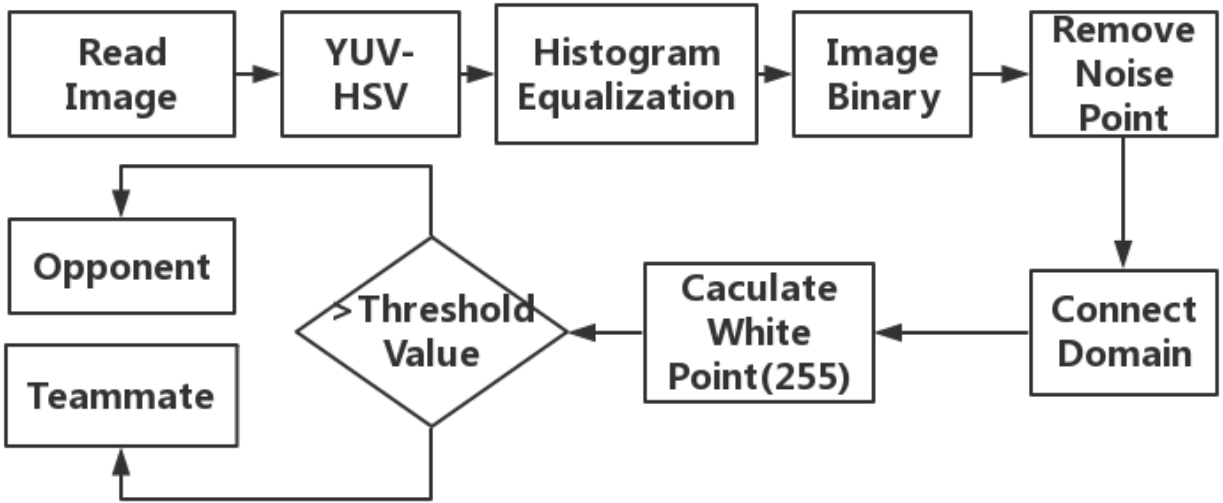


Team Identification

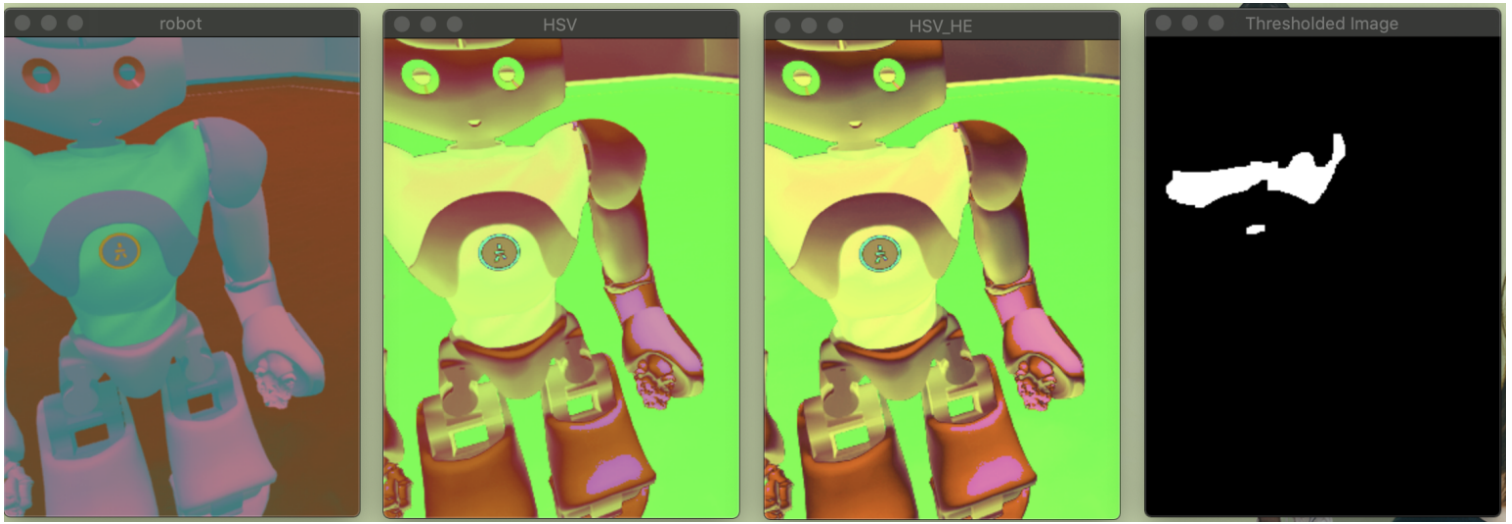
Solutions:

1. Directly use YCrCb

2. Transform YCrCb into HSV



Team Identification



Original Image

HSV Image

HSV after Histogram Equalization

Binary Image

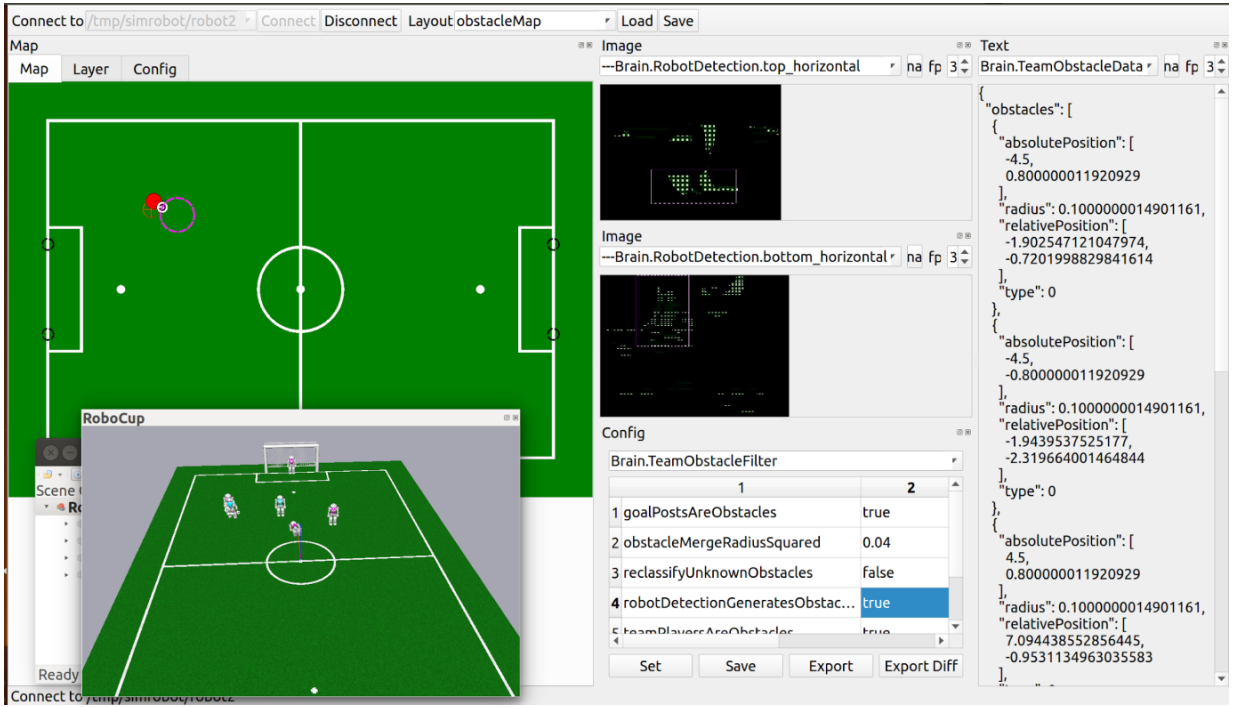
Position Estimate

middle-point of the bottom-line in pixel coordination

Positions of the robots in robot coordination

Pose of the robot in field coordination

Position Estimate



Question?

Future Work and Outlook

Future Work

- More stable purple circle of robot detection in MATE
- Integrate the teammate estimation
- Optimize the memory consumption of the neural network
- Faster the speed of processing

Outlook

- Algorithm which can detect all types of color
- Optimization for reducing the influence of light elements
- More accurate self-positioning

Lehrstuhl für Mustertechnik
Fakultät für Musterverfahren
Technische Universität München

Parameters

Layer(type)	Output Shape	Param #
conv2d	(None, 28, 28, 8)	80
activation	(None, 28, 28, 8)	0
batch_normalization	(None, 28, 28, 8)	32
max_pooling2d	(None, 14, 14, 8)	0
dropout	(None, 14, 14, 8)	0
conv2d_1	(None, 12, 12, 16)	1168
activation_1	(None, 12, 12, 16)	0
batch_normalization_1	(None, 12, 12, 16)	64
max_pooling2d_1	(None, 6, 6,16)	0
dropout_1	(None, 6, 6,16)	0
global_average_pooling2d	(None, 16)	0
dropout_2	(None, 16)	0
dense	(None, 16)	272
activation_2	(None, 16)	0
dropout_3	(None, 16)	0
dense_1	(None, 1)	17
activate_3	(None, 1)	0

Total Params:

1,633

Trainable Params:

1,585

Non-trainable Params: